软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2012,23(2):230−239 [doi: 10.3724/SP.J.1001.2012.04145]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

嵌入式软件体系结构级能耗建模方法*

刘啸滨[1+], 郭 兵[1], 沈 艳[2], 熊 冰[1], 王继禾[1], 伍元胜[1], 刘云本[2]

[1](四川大学 计算机学院,四川 成都 610065)

[2](成都信息工程学院 控制工程学院,四川 成都 610225)

Embedded Software Energy Modeling Method at Architecture Level

LIU Xiao-Bin[1+], GUO Bing[1], SHEN Yan[2], XIONG Bing[1], WANG Ji-He[1], WU Yuan-Sheng[1], LIU Yun-Ben[2]

[1](College of Computer Science, Sichuan University, Chengdu 610065, China)

[2](College of Control Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

+ Corresponding author: E-mail: aly.lxb@163.com

Liu XB, Guo B, Shen Y, Xiong B, Wang JH, Wu YS, Liu YB. Embedded software energy modeling method at architecture level. *Journal of Software*, 2012,23(2):230−239. http://www.jos.org.cn/1000-9825/4145.htm

Abstract: This paper starts at a software architecture level, considers the functional relation between software characteristic quantities and embedded software energy as nonlinear (linear functional relation can be considered as a special nonlinear functional relation). Next, the paper presents an energy model at architecture level by using BP neural network. The energy model measures 5 software characteristic quantities at architecture level and uses BP neural network to fit the functional relation between software characteristic quantities and embedded software energy. Experimental results show that this model is effective.

Key words: embedded software energy; architecture; energy modeling; software characteristic quantity; characteristic quantity measurement; BP neural network

摘 要: 从软件体系结构层面出发,假设体系结构级软件特征量与嵌入式软件能耗之间存在非线性函数关系(线性函数关系可被认为是一种特殊的非线性函数关系),并以此假设为基础,提出了一种利用 BP 神经网络在体系结构级估算软件能耗的模型.能耗模型对5个体系结构级软件特征量进行度量,使用BP神经网络拟合出软件特征量与嵌入式软件能耗的非线性函数关系,并通过实验验证了模型的有效性和假设的合理性.

关键词: 嵌入式软件能耗;体系结构;能耗建模;软件特征量;特征量度量;BP 神经网络

中图法分类号: TP311 **文献标识码**: A

目前,嵌入式系统在信息家电、智能控制、军事电子等领域得到了广泛的应用.截至 2009 年底,全世界嵌入式设备的保有量超过了 40 亿台,且数量继续呈快速增长的势头,每年电力消耗达到 1 000 亿千瓦时以上.在全球倡导低碳经济的背景下,嵌入式系统的能耗是一个日益引起人们关注的热点问题,成为嵌入式系统设计的重要考量因素[1].

* 基金项目: 国家自然科学基金(61073045); 四川杰出青年科技基金(2010JQ0011)
收稿时间: 2011-07-15; 修改时间: 2011-09-06; 定稿时间: 2011-11-14

© 中国科学院软件研究所 http://www.jos.org.cn

软件特征量揭示了软件的某些本质特征,通过分析某些软件特征量与软件能耗的关系,以软件特征量为基础建立能耗模型是嵌入式软件能耗研究领域内的一种合理、有效的研究思路.本文从体系结构层面出发,提取出有效代码行数、构件数目、平均构件接口复杂度、平均路径复杂度、平均构件耦合度这 5 个体系结构级软件特征量,并假设这些软件特征量与软件能耗存在非线性函数关系,以此为基础建立了嵌入式软件体系结构级能耗模型.能耗模型分析了上述软件特征量影响嵌入式软件能耗的过程,给出了具体的度量方法,并使用 BP 神经网络拟合出软件特征量与嵌入式软件能耗的非线性函数关系.通过实验验证,模型预测值与实际能耗值的误差在 20%以内,说明了模型的有效性,同时也证明了体系结构级软件特征量与嵌入式软件能耗之间存在非线性关系的假设是合理的.

1 研究现状

目前,国外一些代表性的嵌入式软件能耗建模的研究成果主要包括:Tan 在文献[2]中提出了一种使用基于特性的宏模型分析、计算软件能耗的方法.该方法以目标处理器能耗模型刻画函数级能耗,然后根据函数刻画整个软件的宏模型,从而对软件的能耗进行评估.Tan 在文献[3]中提出了一种体系结构转换的方法,从体系结构的角度对嵌入式软件的能耗进行优化.但该模型采用线性回归方法,在面对多种输入时模型不具有通用性,另外,模型的准确性不够稳定.Lee 等人在文献[4]中描述了一种基于进程代数的形式化框架,对能源受限的实时系统进行建模与分析.Senn 等人在文献[5]中针对具体的体系结构建模语言 AADL(the architecture analysis and design language)研究了基于 AADL 模型的嵌入式系统的能耗评估精化方法.

国内对于嵌入式软件能耗建模方面的研究工作起步晚于国外,但经过国内一些科研单位同仁多年的刻苦攻关,迄今已取得了一些可喜的成果,一些代表性的工作包括:张滕滕等人在文献[6]中提出了一种基于 CSP (communicating sequential process)进程代数语言的能耗模型.该模型以接口为基本研究对象,并对系统的最大、最小和平均能耗进行了定义,但文章缺乏对该模型有效性的验证.陈丽琼等人在文献[7]中提出了一种能耗时延 Petri 网(ECTPN),对系统的模块、任务、任务间关系、通信协议等进行建模,分析系统的可调度性和能耗约束,并给出启发式算法计算满足时间约束和能耗约束的可行调度.赵霞等人在文献[8]中提出了一种嵌入式操作系统能耗量化分析方法,利用微体系结构能耗模型估算单时钟周期指令能耗,同时提出基于软件功能结构的操作系统内核能耗估算模型,估算分析内核执行路径、服务、例程、原子函数的能耗,发现操作系统内核中显著影响系统能耗的关键软件模块及其特征.

对嵌入式软件功耗的研究已经成为嵌入式研究领域的研究者所关注的焦点,但上述大多数研究仍然停留在指令级、源程序结构级和算法级的能耗分析与估算上,对更高层次的体系结构级的能耗分析与估算的研究还较少.

2 体系结构级能耗建模方法

传统的体系结构级嵌入式能耗建模主要使用线性回归方法,认为软件体系结构级软件特征量与软件能耗之间存在 n 元线性函数关系,其能耗模型如下:

$$E = c_1 + c_2 s_1 + c_3 s_2 + \ldots + c_{n+1} s_n \tag{1}$$

其中,E表示软件能耗,c_1表示软件初始化所消耗的能耗,S_n表示与软件能耗相关的软件特征量的值,c_n表示各个软件特征量的参数[9].使用线性回归方法,虽然可以提高模型的分析速度,但存在诸多的缺陷:理论上,软件高层度量特征与软件能耗的关联关系为线性回归函数关系的假设缺乏确实的依据支持;实践上,二元线性函数模型固然简单,在能耗评估的速度上具有优势,但能耗模型精确度较差,误差也较大,在许多情况下不能满足实际要求.综合以上分析,本文认为体系结构级软件特征量与嵌入式软件能耗之间的关系为非线性函数关系(线性函数关系可以认为是一种特殊的非线性函数关系),通过分析某些体系结构级软件特征量与软件能耗之间的关系,可以得到新的体系结构级能耗模型:

$$E_S = \overline{P} \times T_S = f(C_S) \times T_S = f(Loc, TC, R_C, R_P, CF) \times T_S \tag{2}$$

© 中国科学院软件研究所 http://www.jos.org.cn

其中,E_S表示软件能耗,$\overline{P}$表示软件的平均功耗,T_S表示软件运行的时间,C_S表示某些与软件功耗相关的软件度量特征,f表示软件功耗与软件相关度量特征的非线性函数,本文将采用BP神经网络进行拟合,Loc表示软件有效代码行数,TC表示构件数量,R_C表示平均构件接口复杂度,R_P表示平均路径复杂度,CF表示平均构件耦合度.软件能耗E_S和软件运行时间T_S可以通过相应的软硬件测量工具获得,平均功耗$\overline{P}$可以通过一组基准程序测试获得平均值,也可以通过功耗特征关联模型获得,Loc等 5 个体系结构级软件特征量影响软件能耗的过程和具体度量方法将在下一节进行论述.以软件特征量为基础的体系结构级能耗建模的方法分为如下 5 个步骤:

(1) 假设体系结构级软件特征量与软件能耗之间存在非线性函数关系.

(2) 从体系结构层面分离出与嵌入式软件能耗相关的软件特征量,具体为以下 5 项:有效代码行数、构件数量、平均构件接口复杂度、平均路径复杂度、平均构件耦合度.

(3) 选择合理的度量方法,对有效代码行数、构件数量、平均构件接口复杂度、平均路径复杂度、平均构件耦合度等体系结构特征量进行度量.

(4) 通过在功耗仿真实验平台HMSim运行多组基准程序,测得E_S,T_S的值,根据公式(1),得出软件运行期间的平均功耗$\overline{P}$值:

$$\overline{P} = E_S / T_S \tag{3}$$

(5) 训练 BP 神经网络拟合出非线性关联函数 f.函数 f 的输入值为各个软件体系结构级特征量的度量值,输出为软件平均功耗.

3 软件特征量的度量

本文从体系结构角度出发,提取了有效代码行数、构件数目、平均构件接口复杂度、平均路径复杂度、平均构件耦合度这 5 个软件体系结构级特征量.由于嵌入式软件运行的环境一般为嵌入式实时操作系统,资源受限,要求嵌入式软件必须有更高的能效比,所以嵌入式软件必须高效率地设计,去除冗余,精简构件,合理设计.因此,嵌入式软件在有效代码行数、构件数目、平均构件接口复杂度、平均路径复杂度、平均构件耦合度等方面就有更高的要求,上述 5 个软件体系结构级软件特征量也就更能代表嵌入式软件的特点.本文具体分析了上述软件特征影响平均功耗$\overline{P}$的过程,并说明如何对上述特征进行有效度量.

3.1 有效代码行数

程序代码通过编译转换成硬件能够识别的二进制指令,嵌入式系统按照二进制指令完成各种操作,在执行过程中,微处理器完成取指、译码、取操作数和执行等操作,其中,取指令和取操作数阶段涉及到内存读写访问和总线数据传输,译码阶段涉及到微处理器内部算术运算和逻辑运算,执行阶段根据不同指令类型完成不同的操作,如内存读写、I/O 读写等,而整个过程中,微处理器内部某些寄存器的值也发生对应的改变.上述一系列操作步骤都将触发电路活动,产生相应的电路能耗.可见,程序代码作为机器指令的来源是嵌入式软件能耗产生的本质原因.

通过实验验证,在固定平台上,每一条二进制指令都有一个固定的能耗值.指令执行的条数越多,对应产生的能耗就越大.软件代码在固定编译工具和编译优化环境中产生的二进制代码是固定的.因此,可以认为软件代码行数决定了软件能耗的大小.

软件代码分为有效代码和无效代码两类.有效代码是指通过编译器能够转化为可执行的机器指令的代码,反之则为无效代码.根据测量各类代码操作运行时的能耗变化,得到以下 4 类操作为有效代码:变量计算、变量赋值、函数初始化、函数调用.为了提高有效代码行数度量的精确度,在度量过程中将对具体代码作如下一些变换:

(1) 针对构件中出现选择的情况,根据实际状况设定选择的概率,与选择中的有效代码行数相乘,结果加 1,作为选择情况的有效代码行数.

(2) 针对构件中出现循环的情况,将循环展开成对应的变量赋值和变量运算语句,再计算有效代码行数.

© 中国科学院软件研究所 http://www.jos.org.cn

(3) 针对类中出现继承的情况,在计算子类有效代码行数时,将父类的共有部分加入有效代码行数的统计中.

(4) 针对友元的情况,在计算友元类有效代码行数时,将基类的保护部分和共有部分加入有效代码行数的统计中.

(5) 针对虚函数的情况,只单独计算子类中的有效代码函数.

根据功耗仿真平台 HMSim 对空函数初始化、空函数调用、简单变量运算和简单变量赋值这 4 种操作的能耗测试,简单变量运算所消耗的能耗是简单变量赋值所消耗能耗的 3 倍,函数初始化所消耗的功耗大约是简单变量赋值所消耗能耗的 7 倍,空函数调用所消耗的能耗与简单变量赋值所消耗功耗基本相等.为了方便有效代码行数的度量,将简单变量赋值操作所消耗的能耗作为基准度量单位.

3.2 构件数量

构件作为对象的延伸,是对功能独立的一组模块进行封装后形成的实体.构件的交互通过连接件进行.可以认为,构件数量决定了软件所包含的模块数量.假设每个模块包含的有效代码行数相等,则构件数量越多,模块数量越多,对应的有效代码行数也越多.由此可见,构件数量能够通过影响有效代码行数对软件能耗进行间接影响.对构件数量的度量相对简单,根据软件体系结构设计图即可得到软件构件数量 TC.

3.3 平均构件接口复杂度

构件通过接口与连接件通信,完成控制流和数据流的交互.构件接口越多,表明该构件被其他构件调用或者触发其他构件的概率越大,构件中的有效代码运行也越频繁.有效代码决定软件功耗,可认为构件接口复杂度对软件功耗有间接的影响.

构件接口按照类型分为输入接口和输出接口.其中,输入接口负责接收其他构件发出的控制流和数据流,输出接口负责向其他构件发出数据流和控制流.输入接口越多,意味着构件可重用性越高;输出接口越少,则意味着构件的耦合性越低.合理的构件应该是可重用性高,耦合性少,这要求软件的输入接口多,输出接口少.在文献[10]中已经证明,构件的输入接口影响构件本身的复杂度,使自身结构按 e(e=2.71828)的指数增长,构件的输出接口不仅影响构件本身的复杂度,更重要的是影响其他构件的接口复杂度,并使其本身的接口复杂度按 2 的指数增长.

假设在 m 个构件组成的嵌入式软件中,构件 j 的输入接口复杂度为

$$I_j = m \times \sum_{i=1}^{q} \mathrm{e}^{|l_j - l_i - 1|} \tag{4}$$

其中,I_j为输入接口复杂度,q表示输入接口总数,l_j表示j构件深度层号,l_i表示与i号接口发生交互的构件深度 层号.

构件 j 的输出接口复杂度为

$$O_j = m \times \sum_{i=1}^{p} 2^{|l_j - l_i + 1|} \tag{5}$$

其中,O_j表示输出接口复杂度,p表示输出接口总数.

构件j的接口复杂度R_j可以表示为

$$R_j = \sqrt[3]{I_j + O_j} \tag{6}$$

整个软件的平均接口复杂度R_C可以表示为

$$R_C = \frac{1}{m} \times \sum_{j=1}^{m} R_j \tag{7}$$

3.4 平均路径复杂度

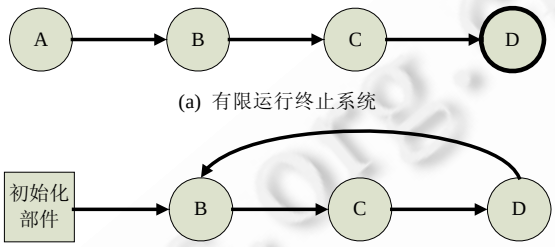
软件顺序执行的过程可以视为软件执行路径,执行过程中参与运行的构件数量可以视为路径的长度,不同的执行过程可以视为软件不同的分支路径.路径长度越长,表示参与运行的构件数量越多,构件数量可以间接影

© 中国科学院软件研究所 http://www.jos.org.cn

响软件能耗,可以认为路径长度对软件能耗有间接的影响.不同的分支路径,路径长度不同,对应的软件能耗也不相同.由此可见,路径复杂度能够通过影响实际运行的构件数量对软件能耗产生间接影响.

路径复杂度的度量根据系统执行的具体流程可以分为两种情况:单一路径复杂度度量和多分支路径复杂度度量.

单一路径指系统只存在 1 条运行路径,各构件按路径顺序执行.嵌入式系统分为有限运行终止和无限循环两种情况,两者的度量方法也存在差异.图 1 为两者示意图:



(a) 有限运行终止系统

(b) 无限循环系统

Fig.1 Diagram of finite running system and infinite loop system

图 1 有限运行终止系统和无限循环系统示意图

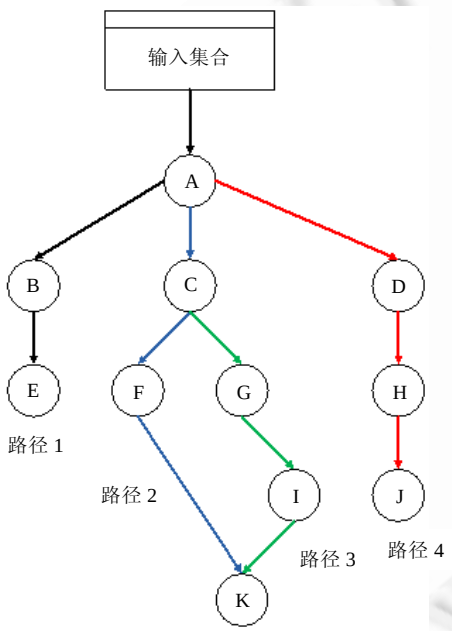


Fig.2 Diagram of multi-branch path system

图 2 多分支路径系统示意图

有限运行终止系统,即系统成功执行完毕后进入终止状态阶段,其路径复杂度等于路径执行过程中执行的构件数目,如在图 1(a)中系统的路径复杂度为 4;无限循环系统,即系统经过一系列初始化部件后进入无限循环状态,在无限循环系统中,系统运行时间越长,初始化部件所产生的能耗在系统总能耗中所占的比例越小,在经过足够长的运行时间后,可以认为初始化部分能耗占系统总能耗的比例趋近于 0,因此,可以认为无限循环系统的路径复杂度等于无限循环部分中的构件数量,如在图 1(b)中,系统的路径复杂度为 3.

多分支路径即系统运行过程包含多条执行路径,系统根据不同的输入集合选择不同的执行路径,不同的执行路径对应有不同的执行概率.多分支路径系统如图 2 所示.

多分支路径的路径复杂度度量情况相对复杂,有如下两种情况需要对路径进行变换:针对多分支路径中存在回路的情况,因为回路的出现会降低软件可靠性,故将回路部分分解成多条路径,每条路径中保证回路部分的构件只执行一次;针对多分支路径中存在无限循环部分的情况,为循环部分添加强制结束节点,分支路径的长度为循环部分所包含的构件数目.

路径复杂度R_P度量公式如下:

$$R_P = \sum_{i=1}^{m} p_i l_i \tag{8}$$

其中,m为系统中存在的运行路径数目,p_i为第i条路径执行概率,l_i为第i路径的路径长度.

根据公式,假设图 2 中每条路径执行概率相等,都为 1/4,则在图 2 所示的系统中,4 条路径的长度依次为 3,4,5,4,那么由公式(8)得到该系统平均路径复杂度为 3×1/4+4×1/4+5×1/4+4×1/4=4.

© 中国科学院软件研究所 http://www.jos.org.cn

3.5 平均构件耦合度

耦合度的概念最早使用在软件工程领域,主要用于衡量一个模块与另一个模块的紧密程度.耦合度表达了软件结构中模块之间相互关联程度,模块间耦合度越高,系统越复杂,对模块的理解、维护和调试的难度也越高.合理的软件应该追求系统的低耦合.

体系结构由软件工程衍生出来,耦合度的概念仍然可以作为构件重要的质量属性,衡量构件间的复杂程度.构件间耦合程度的高低决定了构件接口的复杂度的大小,构件间耦合程度越高,构件的接口复杂度越高.可见,构件间耦合程度能够通过影响构件接口复杂度来间接影响软件功耗.

目前在软件工程领域,针对模块耦合度的度量已经存在许多成熟的度量方法,其中,MOOD 度量方法得到普遍认同.MOOD 度量是一种面向对象的度量方法.这套方法由 Abrito 等人提出,用来度量软件的封装性、继承性、耦合性和多态性.将 MOOD 度量方法运用在构件耦合度的度量上,得到如下构件平均耦合度(CF)的度量公式:

$$CF=\frac{\sum_{i=1}^{TC}\left[\sum_{j=1}^{TC} is_client(C_i,C_j)\right]}{TC^2-TC} \tag{9}$$

其中,TC 表示构件总数,且 $is_client=\begin{cases}1, & C_i\text{与}C_j\text{存在耦合关系}\\ 0, & C_i\text{与}C_j\text{不存在耦合关系}\end{cases}$.

判断两个构件是否存在耦合关系,即是判断两个构件是否存在控制交互或者数据交互,如果存在控制交互或数据交互,则两个构件存在耦合关系;反之,则不存在耦合关系.模块间的耦合度是指模块之间的依赖关系,包括控制关系、调用关系、数据传递关系.也可以通过判断两个构件是否存在上述依赖关系来确定它们之间是否存在耦合关系.

4 BP 神经网络非线性拟合

体系结构软件特征量与嵌入式能耗存在非线性关系,为了更加精确地描述这种非线性关系,选择一种合理有效的数值逼近方法是十分必要的.BP 神经网络作为一种数值逼近方法,在数学上能够逼近任意非线性函数,具有较高的拟合度,因此,本文采用 BP 神经网络对体系结构特征与软件能耗进行拟合.

4.1 BP神经网络非线性函数拟合流程

使用 BP 神经网络实现非线性拟合的具体流程如下:

(1) 对大量样本程序的 5 个软件体系结构级特征量(有效代码行数、构件数量、平均构件接口复杂度、平均路径复杂度、平均构件耦合度)分别进行度量.

(2) 对 5 个软件特征量的度量值进行预处理,处理后的值作为 BP 神经网络的输入值.

(3) 在功耗仿真实验平台 HMSim 上运行样本程序,获得样本程序的能耗值 E,E 将作为 BP 神经网络的输出值.

(4) 确定 BP 神经网络的具体结构,包括隐层数、隐层节点数、隐层传递函数、输出层传递函数等.

(5) 输入样本程序软件特征量度量值和能耗值对 BP 神经网络进行训练,通过训练确定 BP 神经网络隐层的权值和阈值.

(6) 输入目标程序的特征量到训练好的 BP 神经网络中,得到能耗预测值,再与目标程序的实际能耗值进行对比,验证建模方法的有效性.

4.2 确定BP神经网络的结构

在获得有效的体系结构特征度量值后,需要进一步确定 BP 神经网络的结构,需要考虑的因素包括:神经网络学习速率、逼近误差、收敛速度、内存占用情况等.

(1) 隐层数的确定.隐层数多少决定了网络误差,但过多的隐层数会导致网络复杂化,增加网络训练时间和过拟合的倾向.文献[11]中证明了单隐层的 BP 神经网络可以逼近任何在闭区间内的连续函数,一个 3 层的 BP

© 中国科学院软件研究所 http://www.jos.org.cn

网络可完成任意的 n 维到 m 维的映射.因此,隐层数确定为 1.

(2) 隐层节点数的确定.BP神经网络中,隐层节点数对BP网络性能影响巨大,但目前理论上还缺乏科学的和普遍的方法用于隐层节点数的确定.一般采用以下经验公式确定隐层节点数n_1:

$$n_1=\sqrt{n+m}+a \tag{10}$$

其中,n 为输入层节点数,m 为输出层节点数,a 为 1~10 的常数.对应到体系结构级能耗模型中,输入层包括有效代码行数、构件数量、平均构件接口复杂度、平均路径复杂度和平均构件耦合度 5 个输入量,因此,输入层节点数为 5,输出层为平均功耗 $\overline{p}$, 输出层节点数为 1.因此,n_1的取值范围为 3~12.

(3) 各层函数的确定.针对单隐层 BP 神经网络,输入层和隐层传递函数一般采用 tansig 函数,输出层传递函数一般采用 purelin 函数.通过多组实验可以发现,Trainlm 函数在隐层节点数为 11 时,其收敛速度和逼近误差都能达到令人满意的效果.因此,BP 神经网络的训练函数使用 Trainglm,训练函数的隐层节点数为 11.

在确定各个参数以后,可以开始进行网络训练.通过输入大量样本数据,在最小均方误差精度为 10^{-7}的情况下,得到BP神经网络的权值w_1,w_2,和阈值b_1,b_2,如图 3 所示.

$$w_1=\begin{bmatrix} -0.4165 & -0.1696 & -1.4500 & -0.0442 & 0.4560 \\ 1.5794 & -1.1164 & -3.6486 & -0.1652 & 3.1644 \\ -9.8318 & 1.4446 & 1.7938 & 1.2356 & -3.6989 \\ 2.3582 & 2.2668 & 1.4666 & 1.5912 & 1.0127 \\ -5.7847 & 3.0148 & 1.1429 & -1.8104 & -0.0461 \\ -0.5514 & -2.2003 & 1.3560 & 1.0117 & -1.2311 \\ 2.0483 & -0.6820 & 2.9545 & 1.5495 & -0.6761 \\ 2.3434 & -1.3524 & 2.1790 & -1.5782 & -1.5216 \\ 5.0697 & -2.8251 & 0.6950 & -1.2708 & -2.5102 \\ -10.4356 & 3.6496 & -0.1175 & 2.3541 & 2.7753 \\ -0.8329 & -0.3859 & -3.5933 & 2.2392 & -0.6455 \end{bmatrix},\ b_1=\begin{bmatrix} 1.0532 \\ -3.0581 \\ -0.2066 \\ -2.4258 \\ -3.6183 \\ -0.6104 \\ 0.0659 \\ -0.0099 \\ 3.8499 \\ -7.1203 \\ 0.1079 \end{bmatrix},\ w_2=\begin{bmatrix} -1.5049 \\ -2.8917 \\ -2.0691 \\ -2.1976 \\ -0.1864 \\ -0.2584 \\ -0.4635 \\ 0.0916 \\ -5.8476 \\ -2.1607 \\ 0.0501 \end{bmatrix},\ b_1=0.9315$$

Fig.3 Weight and threshold value of BP neural network

图 3 BP 神经网络权值与阈值

5 实验验证与实验分析

为了验证基于软件特征量的嵌入式软件能耗模型的有效性,本文采用 3 组总共 27 个目标程序对模型进行验证,这些程序中包含了较为常用的和较为复杂的 MP3 文件解码程序.本文中能耗模型的实验环境中测试平台为 HMSim 功耗仿真平台.该平台是一款指令级高精度嵌入式系统功耗模拟器.它不仅能够从功能上完整地模拟一台典型的嵌入式硬件系统(包括 CPU、存储器、时钟中断、I/O、LCD 等),使嵌入式软件能够在 HMSim 上运行,并得到正确的结果,而且还能在模拟软件运行的同时准确地统计软件运行所消耗的能耗.27 个目标程序的有效代码行数、构件数量、平均构件接口复杂度、平均路径复杂度和平均构件耦合度这 5 个软件特征量的具体数值见表 1~表 3.

Table 1 Characteristic quantity values of target programs in Group 1

表 1 第 1 组目标程序软件特征量值

Loc	82	91	284	225	172	149	330	197	421	467
TC	3	3	4	4	4	4	5	5	7	7
Rc	2.043 85	2.043 85	2.965 98	2.965 98	2.719 82	2.719 82	3.191 56	2.276 48	2.899 5	2.662 01
Rp	4	4	11	8	6	6	11	6	13	9
CF	0.5	0.5	0.5	0.5	0.416 67	0.416 67	0.35	0.25	0.214 29	0.190 48

© 中国科学院软件研究所 http://www.jos.org.cn

Table 2 Characteristic quantity values of target programs in Group 2

表 2 第 2 组目标程序软件特征量值

Loc	452	710	319	510	526	641	1437	1902	355	2027
TC	7	8	4	8	8	9	11	14	4	22
Rc	2.662 01	2.783 12	2.965 98	2.897 15	2.983 12	3.096 51	3.102 21	2.719 82	2.719 82	4.009 93
Rp	9	10	8	13	10	12	18	21	6	31
CF	0.190 48	0.156 25	0.5	0.2	0.25	0.148 15	0.248 25	0.202 17	0.416 67	0.104 51

Table 3 Characteristic quantity values of target programs in Group 3

表 3 第 3 组目标程序软件特征量值

Loc	99	212	301	230	468	152	1072
TC	3	5	6	4	6	3	10
Rc	2.043 85	3.191 56	2.276 48	2.880 31	2.719 82	2.043 85	2.710 44
Rp	4	8	13	9	12	3	21
CF	0.5	0.32	0.361 11	0.562 5	0.340 12	0.333 33	0.248 25

本文搭建了一个使用 ARM7TDMI 处理器和μ COS-Ⅱ系统的实验环境进行仿真验证.验证结果如图 4 所示.

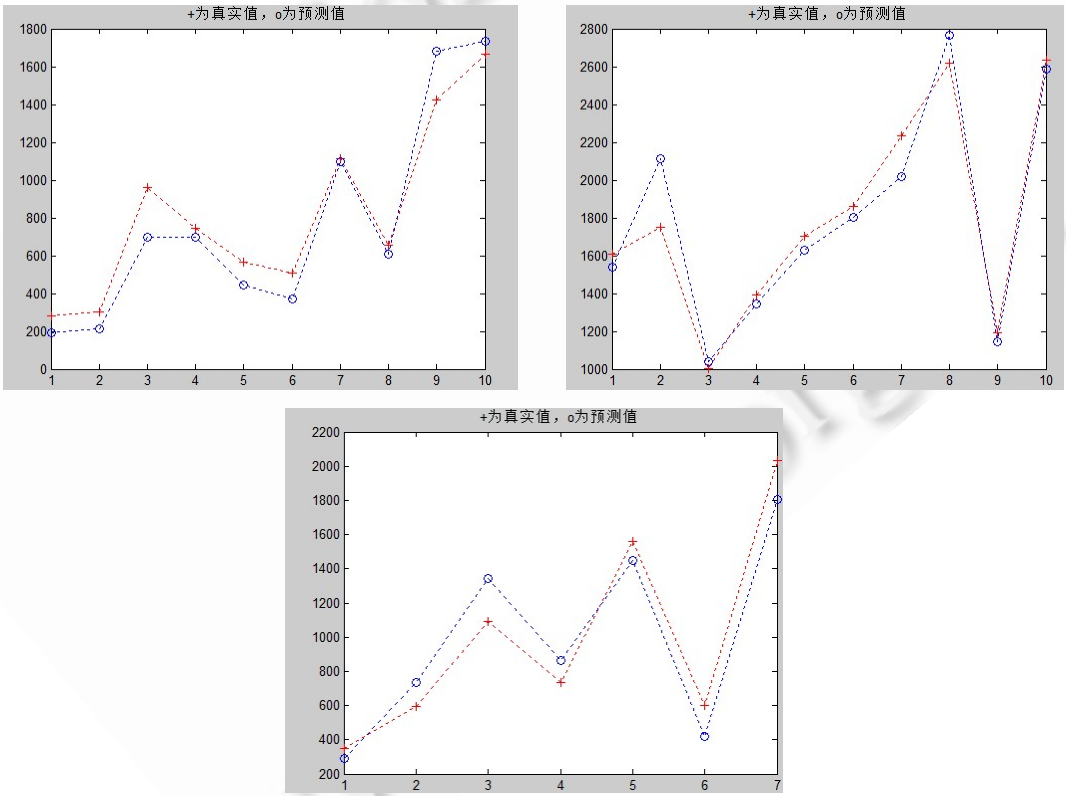


Fig.4 Comparison between three groups of target data and predicted value of model

图 4 3 组目标数据与模型预测值对比

其中,第 1 组目标数据的模型预测的能耗值与实际能耗值的最大误差为 33.4%,最小误差为 3.1%,其平均误差为 17.6%;第 2 组目标数据的模型预测值和实际能耗值的最大误差为 17%,最小误差为 1.2%,平均误差为 7.4%;第 3 组目标数据的模型预测值与实际能耗值最大误差为 31.7%,最小误差为 4.9%,平均误差为 13.4%.

结合表 1 中的数据和图 4 不难看出,能耗值与每个特征量之间存在一定的相关性,最为明显的为特征量

© 中国科学院软件研究所 http://www.jos.org.cn

LOC,即有效代码行数.表 1 中 LOC 较大的目标程序,一般能耗值也较大,也就是图 4 中对应点走势较高,但也有例外.当然,能耗值并不只与 LOC 相关,还要受到其他特征量的影响,这也就是上述例外情况出现的原因.除 LOC 外的其余特征量与能耗值的关系并不能从图 4 中轻易看出,可见每个特征量对能耗值的影响程度存在较大的差异.

通过对实验数据进行分析,可以得出以下几点结论:

(1) 以软件特征量为基础的体系结构级能耗建模方法是有效的.通过 3 组实验数据的对比可以发现,模型的平均误差保持在 20%的范围内,对于处于软件体系中最高层次的软件体系结构来说,误差在可以接受的范围之内.

(2) 证明了 5 个软件体系结构级特征量(有效代码行数、构件数、构件平均接口复杂度、平均路径复杂度和构件平均耦合度)与嵌入式软件能耗之间存在非线性关系的假设的合理性.

(3) 验证了 5 个软件体系结构级特征量的度量方法是有效的.说明有效代码行数、构件数、构件平均接口复杂度、平均路径复杂度和构件平均耦合度可以作为软件体系结构级分析软件能耗的性能指标.

(4) BP 神经网络作为函数逼近工具,在以软件特征量为基础的软件体系结构能耗建模过程中,能够有效地反映软件特征量与软件能耗的非线性关系.

(5) 能耗值与每个特征量之间存在一定的相关性,每个特征量都会对能耗产生一定的影响,但影响程度存在较大的差异.

6 总 结

本文从软件体系结构层面出发,提出软件体系结构级特征量与软件能耗之间存在非线性函数关系的假设,在此基础上,提取出有效代码行数、构件数、构件平均接口复杂度、平均路径复杂度和构件平均耦合度这 5 个软件体系结构级特征量,分析其影响软件能耗的过程并对其进行度量,以 BP 神经网络作为函数拟合手段建立了体系结构级能耗模型,通过实验验证了模型的有效性和假设的合理性.目前在优化方法研究方面也做了一定的研究工作,通过构件合并、构件拆分、构件替换和体系结构风格变化这 4 种软件体系结构级能耗优化方法进行合理组合应用.经过一定的实验验证,实验结果表明,该方法能够有效地降低嵌入式软件能耗,优化比例达到 5%~36%,平均优化比例为 24%.下一步的工作主要是以 5 个软件特征量为基础,在软件体系结构级能耗优化方面做更深一步的研究探讨.

致谢 在此,我们向对本文的工作给予支持和贡献的同学和老师,尤其是向四川大学郭兵教授和他的研究生们表示感谢.

References:

[1] Guo B, Shen Y, Shao ZL. The redefinition and some discussion of green computing. Chinese Journal of Computers, 2009,32(12): 2311−2319 (in Chinese with English abstract).

[2] Tan TK, Raghunathan AK, Lakishminarayana G, Jha NK. High-Level software energy macro-modeling. In: Proc. of the 38th ACM Conf. on Design Automation. 2001. 605−610. [doi: 10.1109/DAC.2001.156211]

[3] Tan TK, Raghunathan AK, Jha NK. Software architectural transformations: A new approach to low energy embedded software. In: Proc. of the Design Automation Test in Europe. 2003. 1046−1051. [doi: 10.1109/DATE.2003.1253742]

[4] Lee I, Philippou A, Sokolsky O. Process algebraic modelling and analysis of power-aware real-time systems. Journal of Computing and Control Engineering, 2002,13(4):180−188.

[5] Senn E, Laurent J, Juin E, Diguet JP. Refining power consumption estimations in the component based AADL design flow. In: Proc. of the IEEE Conf. on Specification, Verification and Design Language. 2008. 173−178. [doi: 10.1109/FDL.2008.4641441]

[6] Zhang TT, Wu X, Li CD, Dong YW. On energy-consumption analysis and evaluation for component-based embedded system with CSP. Chinese Journal of Computers, 2009,32(9):1−8 (in Chinese with English abstract).

© 中国科学院软件研究所 http://www.jos.org.cn

[7] Chen LQ, Shao ZQ, Fan GS. Energy consumption modeling and analysis for distributed real-time and embedded systems. Journal of East China University of Science and Technology (Natural Science Edition), 2009,35(2):250−255 (in Chinese with English abstract).

[8] Zhao X, Guo Y, Lei ZY, Chen XQ. Estimation and analysis of embedded operating system energy consumption. Acta Electronica Sinica, 2008,36(2):209−215 (in Chinese with English abstract).

[9] Seo CY, Malek S, Medvidovic N. Estimating the energy consumption in pervasive Java-based systems. In: Proc. of the 7th Working IEEE/IFIP Conf. on Software Architecture (WICSA 2008). New York: IEEE Press, 2008. 277−280. [doi: 10.1109/PERCOM.2008.85]

[10] Wang XM. A model of measuring software structure complexity and its auto-realizaton. Computer Application, 2009,19(6):16−19 (in Chinese with English abstract)

[11] Sarajedini A, Hecht-Nielsen R. The best of both worlds: Casasent networks integrate multilayer perceptrons and radial basis functions. In: Proc. of the IEEE Int'l Joint Conf. on Neural Networks. 1992. 905−910. [doi: 10.1109/IJCNN.1992.227084]

附中文参考文献:

[1] 郭兵,沈艳,邵子立.绿色计算的重定义与若干探讨.计算机学报,2009,32(12):2311−2319.

[6] 张滕滕,吴晓,李长德,董云卫.基于 CSP 的构件化嵌入式软件能耗分析与评估方法研究.计算机学报,2009,32(9):1−8.

[7] 陈丽琼,邵志清,范贵生.分布式实时嵌入式系统的能耗建模与分析.华东理工大学学报(自然科学版),2009,35(2):250−255.

[8] 赵霞,郭耀,雷志勇,陈向群.基于模拟器的嵌入式操作系统能耗估算和分析.电子学报,2008,36(2):209−215.

[10] 王小明.一种软件结构复杂度度量模型及其自动实现.计算机应用,1999,19(6):16−19.

刘啸滨(1987－),男,河北唐山人,博士生,CCF 学生会员,主要研究领域为嵌入式实时系统.

郭兵(1970－),男,博士,教授,CCF 高级会员,主要研究领域为 RTOS,嵌入式软件开发,SoC 芯片.

沈艳(1973－),女,博士,副教授,主要研究领域为智能化网络化测控技术,智能仪器.

熊冰(1985－),男,硕士,主要研究领域为嵌入式实时系统.

王继禾(1986－),男,博士生,CCF 学生会员,主要研究领域为嵌入式实时系统.

伍元胜(1986－),男,博士生,CCF 学生会员,主要研究领域为嵌入式实时系统.

刘云本(1981－),男,博士生,主要研究领域为嵌入式实时系统

© 中国科学院软件研究所 http://www.jos.org.cn